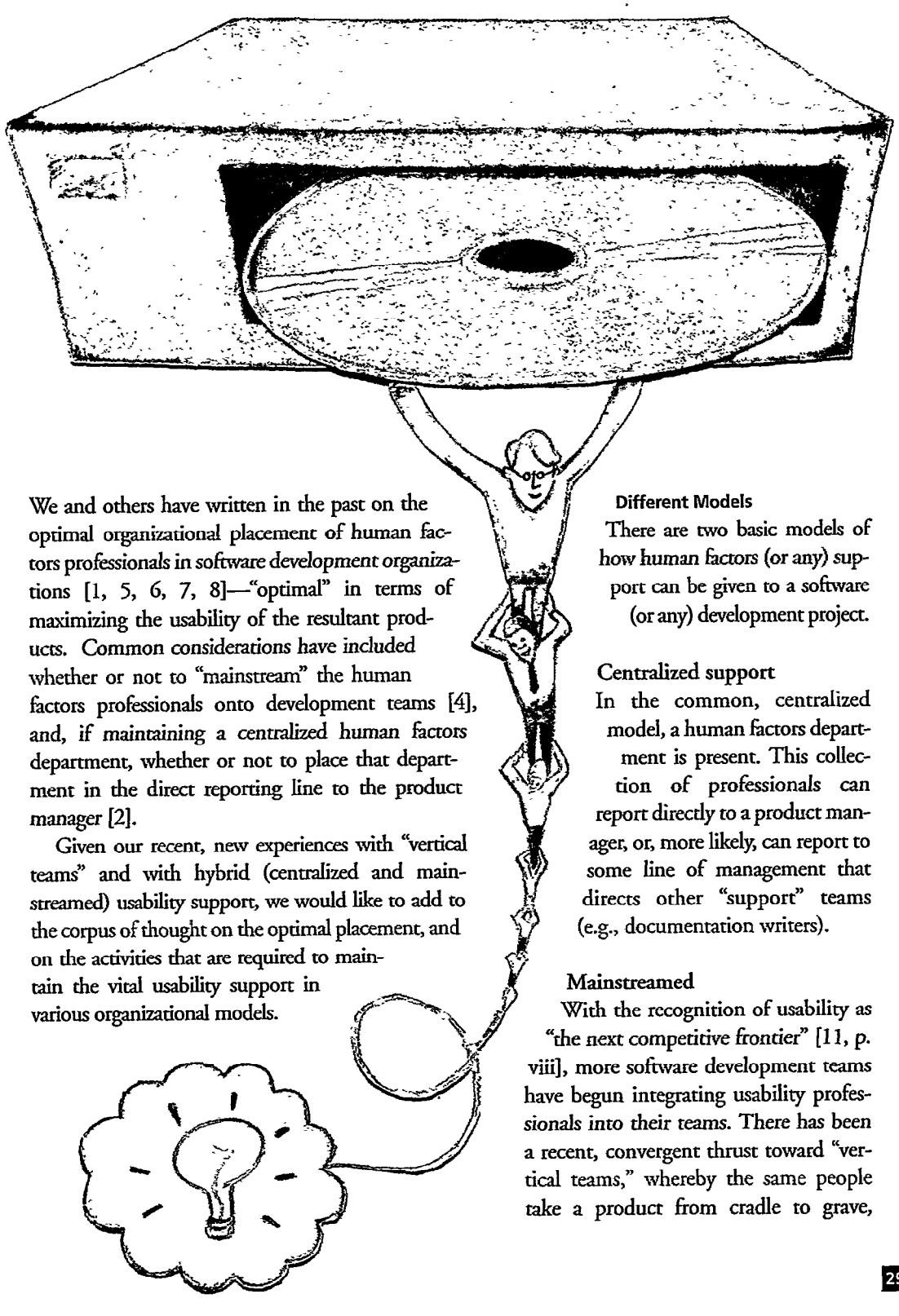
RANDOLPH G. BIAS and PETER B. REITMEYER

Usability Support Inside and Out



We and others have written in the past on the optimal organizational placement of human factors professionals in software development organizations [1, 5, 6, 7, 8]—"optimal" in terms of maximizing the usability of the resultant products. Common considerations have included whether or not to "mainstream" the human factors professionals onto development teams [4], and, if maintaining a centralized human factors department, whether or not to place that department in the direct reporting line to the product manager [2].

Given our recent, new experiences with "vertical teams" and with hybrid (centralized and mainstreamed) usability support, we would like to add to the corpus of thought on the optimal placement, and on the activities that are required to maintain the vital usability support in various organizational models.

all illustrations Ana Cica

Different Models

There are two basic models of how human factors (or any) support can be given to a software (or any) development project.

Centralized support

In the common, centralized model, a human factors department is present. This collection of professionals can report directly to a product manager, or, more likely, can report to some line of management that directs other "support" teams (e.g., documentation writers).

Mainstreamed

With the recognition of usability as "the next competitive frontier" [11, p. viii], more software development teams have begun integrating usability professionals into their teams. There has been a recent, convergent thrust toward "vertical teams," whereby the same people take a product from cradle to grave,

doing the planning, design, development, testing, and delivery.

A Hybrid Model — Usability Support Inside and Out

We at IBM-Austin recently performed some software development while employing a hybrid model. Some of the human factors support came from professionals centralized in a human factors department. Other human factors practitioners were mainstreamed onto product development teams.

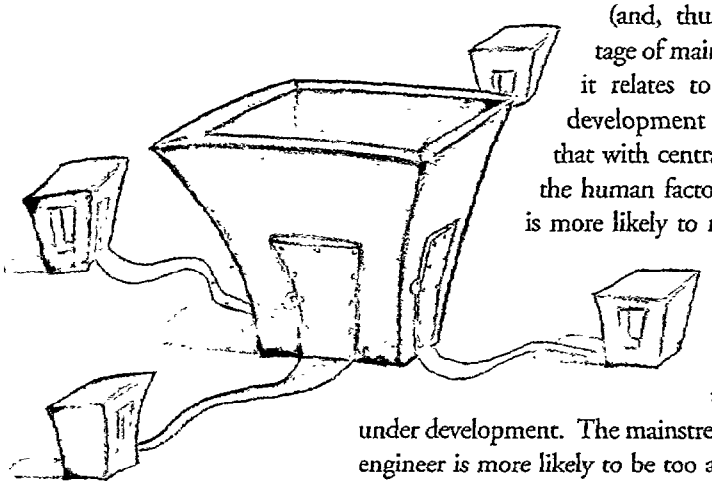
Advantages and Disadvantages

Within any product development organization there should be two distinct objectives (from a usability standpoint): to produce usable products, and to maintain (and grow) usability engineering expertise. Each of the aforementioned organizational models has its own advantages and disadvantages in the pursuit of these two objectives.

Randolph G. Bias and Peter B. Reitmeyer
IBM Corporation
Austin, TX 78758
bias@ausvm1.vnet.ibm.com

The Centralized Model over Mainstreaming

One primary advantage of the centralized model (and, thus, a disadvantage of mainstreaming), as it relates to the product-development objective, is that with centralized support, the human factors professional is more likely to maintain some distance from, and thus objectivity toward, the product under development. The mainstreamed usability engineer is more likely to be too aware of development constraints, and thus not "aim high enough" when it comes to usability objectives.



Regarding the objective of maintaining usability engineering expertise, one primary advantage of the centralized model is it facilitates maintaining a human factors "critical mass." The "care-and-feeding" of the human factors professional is very likely to be monitored, and good, under such a model. Relatedly, it is easier to hire new human factors professionals into such an organization, and the department manager is likely to know how to evaluate the professional's contribution. There is danger, under the mainstreamed model, that a manager unaware of the value of usability will try to deploy a human factors professional as a coder, writer, vendor liaison, or in other ways lose the technical usability vitality. Indeed, the fact of mainstreaming makes harder the interchange among human factors professionals, and calls for more intentionality about maintaining technical vitality. Finally, a centralized human factors department helps with resource allocation balancing; when one product reaches a point in the development process where relatively little usability support is needed, human factors practitioners can be shunted to other projects.

Mainstreaming over the Centralized Model

The beauty of being a mainstreamed human factors professional is being, and being seen as, a team member. The clear buy-in of someone in the development group per se helps with communications to and from the rest of the product developers. For the centralized human factors professional, there is a constant need to insinuate oneself into design decisions, with much less of such a need when serving directly on the development team. Plus, and relatedly, the mainstreaming model tends to allow a much crisper understanding of the product on the part of the human factors professional.

The mainstreamed model can actually *help* with the spread of usability education, as each developer who enjoys a human factors professional as a teammate has the opportunity to learn first-hand the techniques of usability engineering.

Overcoming the Disadvantages

There are actions that can be taken to overcome the disadvantages mentioned above. In the Centralized model, most of the problems of "buy-in" and being seen as a team member can be overcome by cohabitation with the development team being supported, and "dotted-line reporting" to that team manager.

In the mainstreamed realm, with the main problem being maintenance of the human factors community, communications is the main problem. When working with totally mainstreamed human factors support in years past, we established an on-line human factors conference disk and forum, to facilitate intra-discipline communications. Of additional help would be

apparent and actual upper management support of human factors, to encourage the other developers to embrace the usability experts as part of the development team. Further, education of the management team about the role and value of human factors should help avoid the misuse of that resource. Consummately, management willingness to share human factors resource across departments/organizations will yield greater returns for the entire business.

The Special Case of Hybrid Support

Indeed ALL organizations should be, ultimately, hybrid — with all developers becoming usability advocates themselves, yet with some centralized source of usability resource. In one recent example, "Finally, the biggest payoff from the integration of user interface design processes into the software development cycle may come from the fact that now the entire product team is responsible for the user interface" [10].

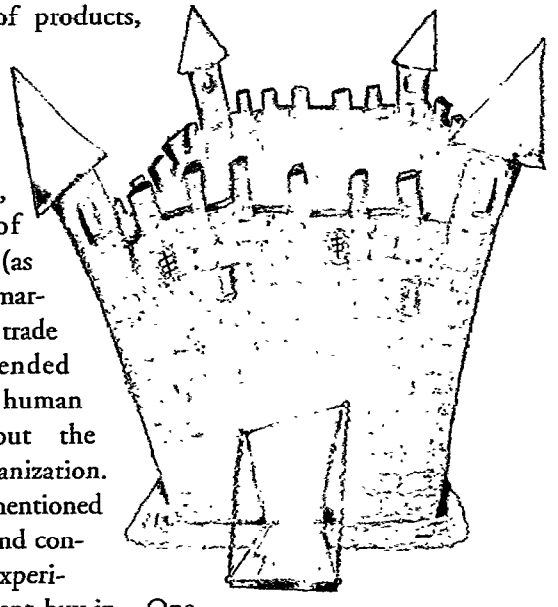
One particular advantage of a hybrid model is good usability coverage; the centralized human factors professionals can "check the work of" the mainstreamed folks. That is, a new, more objective look can be given products that have been developed with mainstreamed usability support.

Of course, this creates a new disadvantage; a new us-vs.-them between the two types of human factors professionals. This is on top of the extant dichotomy between coders and human factors professionals. One way this potential disadvantage can be treated is by rotating human factors professionals into and out of the centralized group. Managers may balk at losing training investment, but the big-picture pay-off will be positive for both the products and the human factors professionals.

This points to another issue, too big to be addressed in detail here. The hybrid model implies, but does *not* necessarily entail, *additional* human factors resources (i.e., people). When evaluating the addition of personnel, there are cost-benefit analysis techniques (e.g., [3]) that can be employed when cost-justifying the application of any human factors resource.

Discussion

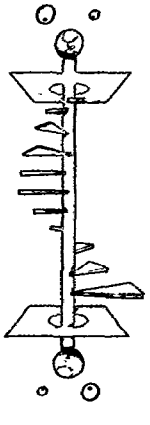
These concerns can be viewed in the context of "understanding and improving software processes... Once a rigorous process architecture is defined, it can be used to guide development, provide a common basis for communication among engineers and managers, and (perhaps most importantly) provide a basis for further process improvement."[9]



For one set of products, developed at IBM-Austin, a hybrid organization was used, with good effects, both in terms of high usability (as reflected in the marketplace and the trade press) and extended appreciation of human factors throughout the development organization. One key factor mentioned in the literature, and confirmed in our experience, is management buy-in. One internal IBM study showed that upper management buy-in was the variable that correlated highest with product usability.

Do we want to become obsolete, and have all usability support done by the developers/coders themselves? No. There is a difference between usability advocacy and usability engineering. And while some of the usability engineering methods will be employed by other developers, some require "a professional."

What to do 'til the doctor comes? If you have no usability resources, gain access to some. The aforementioned cost-justification techniques will allow you to determine if such expenditures are likely to reap rewards. If you do have usability professionals in your organization, consider how they are deployed. Are your products achieving maximal usability, while your technical usability vitality continues to grow? If not, perhaps some of the foregoing suggestions will help.



References

[1] Bias, R. G. (1992). Top 10 ways to muck up an interface project. *IEEE Software*, 9, 95–96.

[2] Bias, R. G., & Alford, J. A. Jr. (1989). Factoring human factoring, in IBM. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp 1296–1300.

BUSINESS COLUMN EDITOR
Susan Dray
Dray & Associates
2115 Kenwood Parkway
Minneapolis, MN 55405
+1 (612) 377-1980
fax: +1 (612) 377-0363
dray.chi@xerox.com
or *sdray@mr.net*

[3] Bias, R. G., & Mayhew, D. J. (Eds). (1994). *Cost-Justifying Usability.* Academic Press: Boston.

[4] Bias, R. G., & Smith-Kerker, P. L. (1986). The mainstreamed human factors psychologist in the development of the IBM RT PC. *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics,* 153–158.

[5] Gillan, D. J., and Bias, R. G. (1992). The interface between human factors and design. *Proceedings of the Human Factors Society Annual Meeting,* pp. 443–447.

[6] Grudin, J. (1991). Systematic sources of suboptimal interface design in large product development organizations. *Human-Computer Interaction,* 6, pp. 147–196.

[7] Grudin, J., Carroll, J., Ehrlich, S., Grisham, M., Hersh, H., & Seybold, P. (1988). Integrating human factors and software development. *Proceedings of CHI 1988: Human factors in computing systems (Washington, D.C., May 15-19).* pp. 157–159. New York: ACM.

[8] Grudin, J., & Poltrock, S.E. (1989). User interface design in large corporations: Coordination and communication across disciplines. *Proceedings of CHI 1989: Human factors in computing systems (Austin, TX, April 30 - May 4).* pp. 197–203. New York: ACM.

[9] Hefley, W. E., Buie, E. A., Lynch, G. F., Muller, M. J., Hoecker, D. G., Carter, J., Roth, J. T. (1994). Integrating human factors with software engineering practices. *Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting,* pp. 315–319.

[10] Miller, M. A., & Stimart, R. P. (1994). The user interface design process: The good, the bad, and we did what we could in two week. *Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting,* pp. 305–309.

[11] Norman, D. A. (1988). *The Design of Everyday Things.* Doubleday: New York.